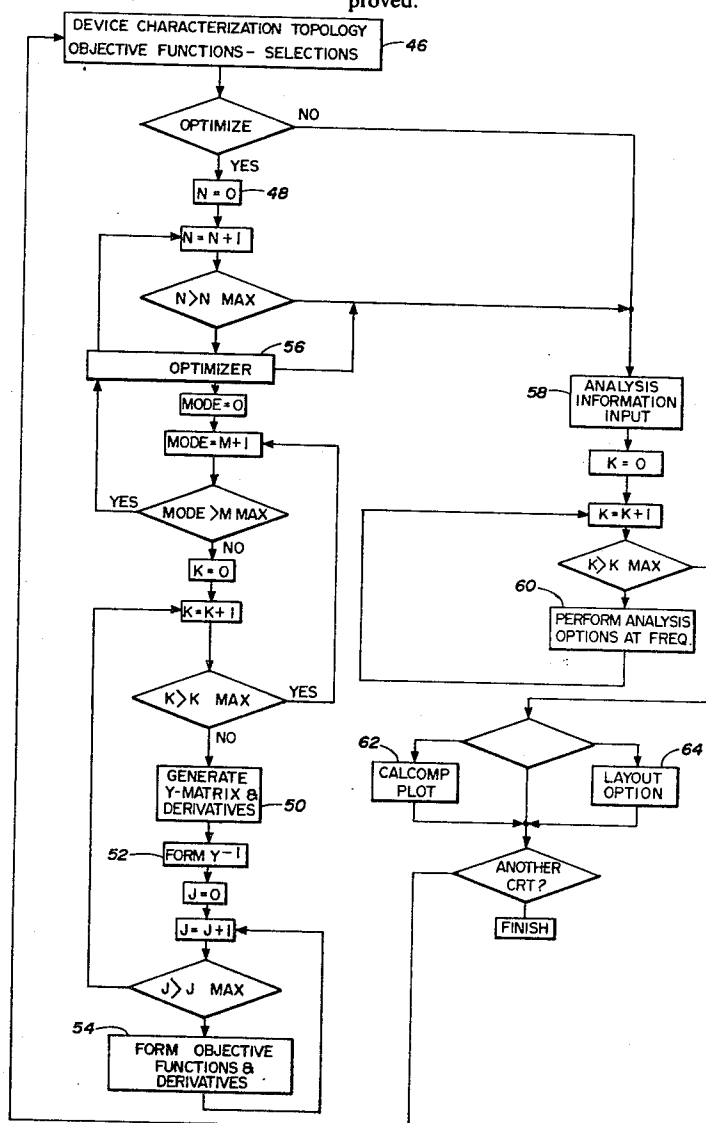


- "Optimal Design of Matching Networks for Microwave Transistor Amplifiers," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-14, pp. 696-698, Dec. 1966; F. E. Emery and M. O'Hagan

Attorneys—James O. Dixon, Andrew M. Hassell, Harold Levine, Melvin Sharp, Henry T. Olsen, Michael A. Sileo, Jr. and John E. Vandigriff

ABSTRACT: Using representations of circuit topologies and initial values of all elements, a circuit design may be modified by an automated data processing machine to improve operation for at least one performance characteristic by changing design variables. A desired circuit topology is selected and the initial values for all the circuit elements are coded and stored in the data processor. In addition, the desired performance characteristic is also selected and coded. Using coded input information, the data processor generates and stores a representation of a continuous analytical objective function. In the operation of the data processor, a value of the objective function is generated and stored. This value is dependent upon the sum of a power function of the difference between an actual performance and a specified performance characteristic. To obtain a specified performance characteristic for a given circuit topology, the value of the objective function is improved.



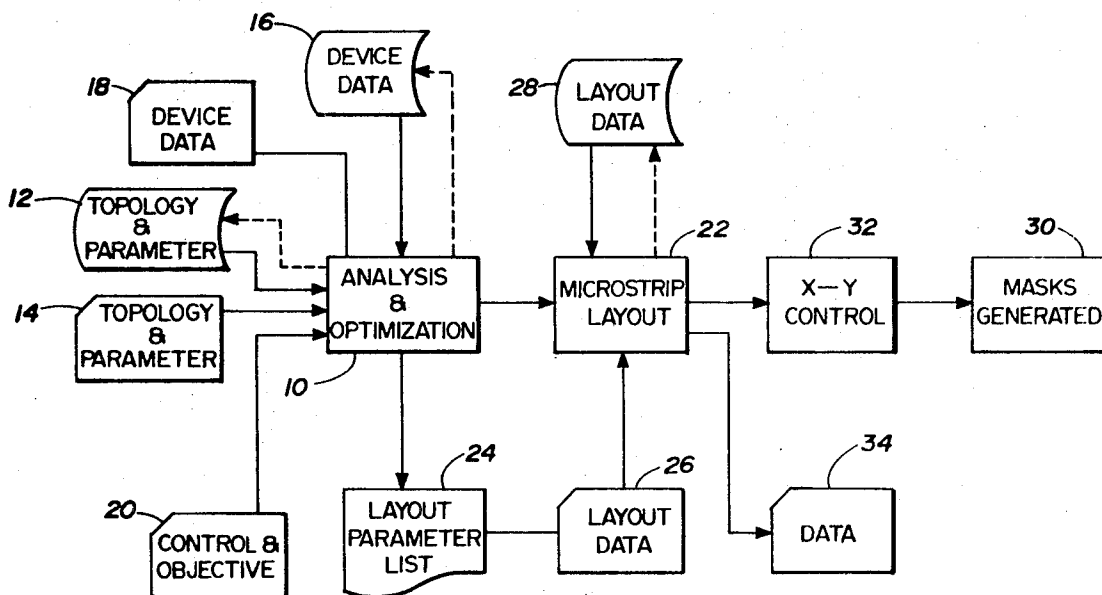


FIG. 1

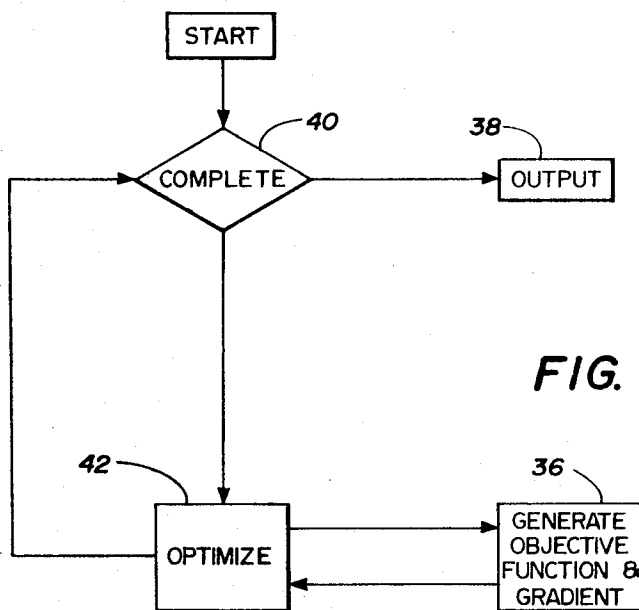
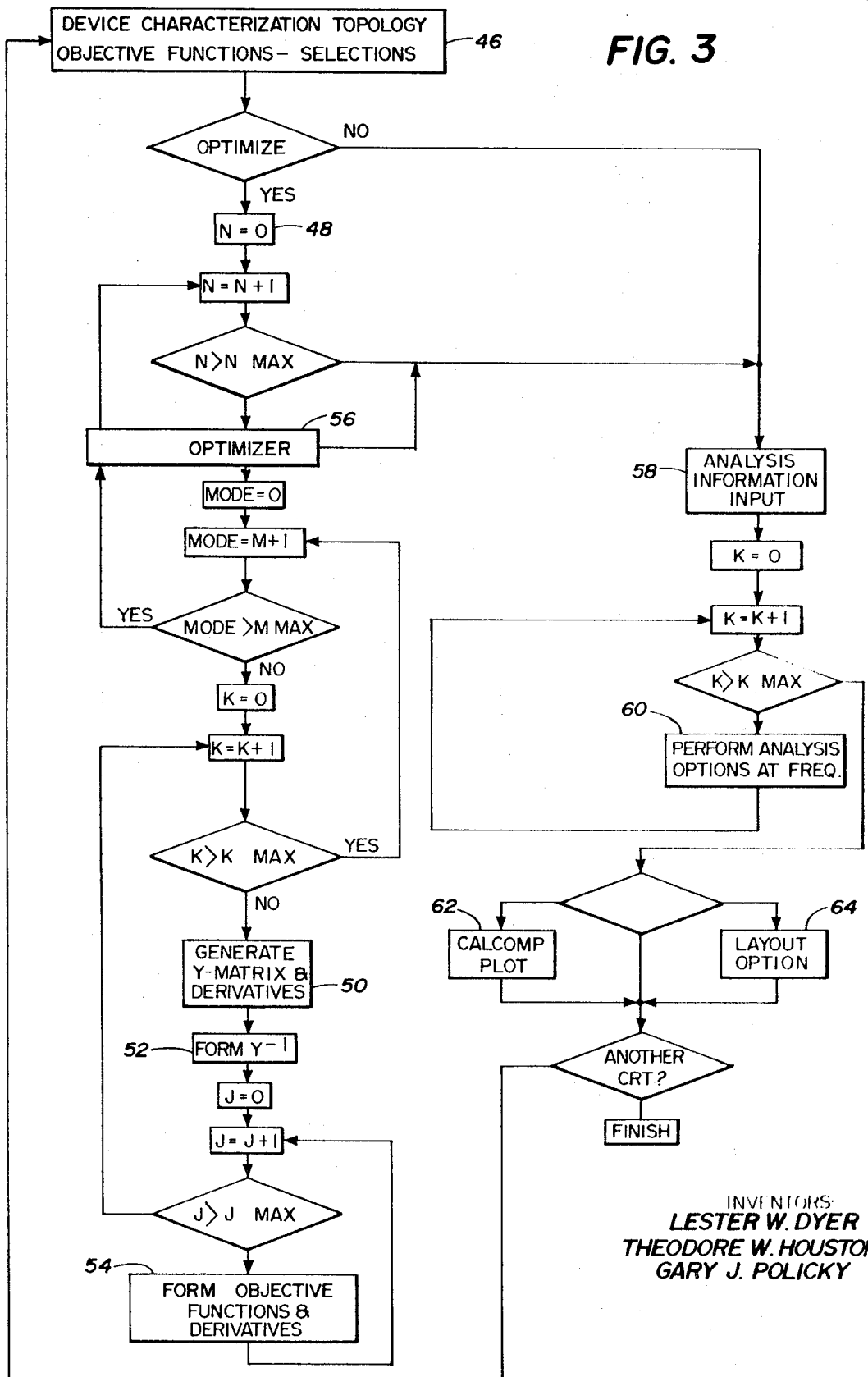


FIG. 2

FIG. 3



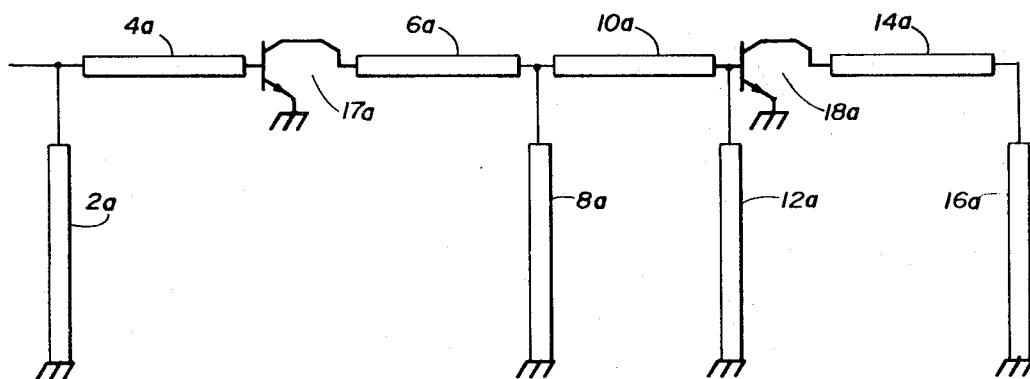


FIG. 4

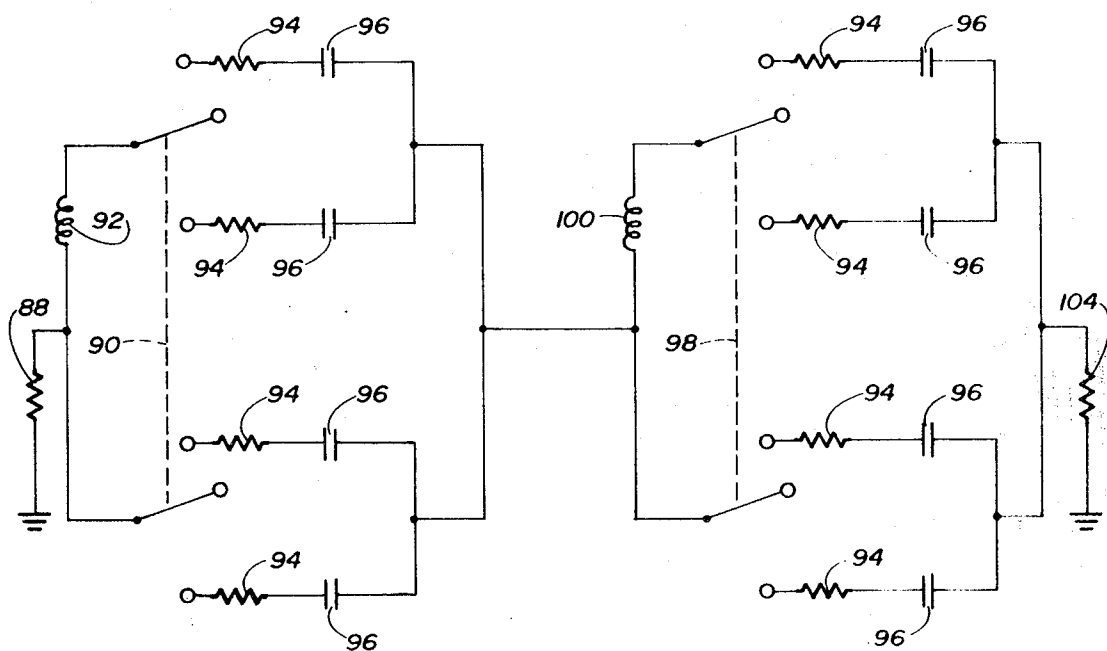


FIG. 5

INVENTORS:
LESTER W. DYER
THEODORE W. HOUSTON
GARY J. POLICKY

CIRCUIT DESIGN BY AN AUTOMATED DATA PROCESSING MACHINE

This invention relates to circuit design, and more particularly to optimization of a circuit design by means of an automated data processing machine.

Circuit designing for years has been the task of an engineer calculating various circuit parameters with a slide rule and paper and pencil. Typically, an engineer is assigned the task of designing a circuit having certain performance characteristics. Either from memory or from a reference text, he selects a circuit topology having known performance characteristics similar to those desired. The designer then calculates the value of each circuit element for a specific performance characteristic. The end result is a circuit that is intended to operate in accordance with the assigned performance characteristic. The final circuit, however, is usually a compromise from what is desired.

In recent years, considerable emphasis has been placed on computerized designs in all fields including electronic circuits, and, as expected, in many cases the computer has outperformed the engineer. It must be recognized, however, that a computer is only a tool in the hands of the design engineer replacing his slide rule. Thus, a computer-designed circuit may be no better than that produced by an engineer using a slide rule. For example, a wide band linear amplifier consisting of several stages might be designed by considering each circuit element of each stage individually and then cascading the results to obtain the desired overall effect. This, in essence, is the same procedure used by the engineer in his manual computations. The desired band width might be achieved by either stagger-tuning or single-tuning the stages. Whether the end result is an amplifier having stagger-tuning or single-tuning depends on the instructions to the computer. In effect, the computer merely relieves the engineer of his slide rule calculations. The final circuit design may operate no better than that designed by hand calculations.

In accordance with the present invention, an engineer selects a representative circuit topology and assigns initial values to all the circuit elements. This information is coded into a computer along with selected options that outline a specified circuit performance characteristic. The automated data processing machine, using a program listing, then considers the overall circuit topology in light of the desired performance characteristic and provides an optimized circuit design. Using the program listing, the data processor changes design variables, including circuit elements, in a manner that is not limited to any specific design approach. The process considers the entire circuit as a whole, and not individual parts that must later be combined. Consider the previous example, all the design variables of the several stages are evaluated simultaneously to improve the performance as specified by selected performance characteristics. The final circuit design may be stagger-tuned, singled-tuned, or some combination. A data processor operating in accordance with the present invention is not limited to any specified approach. Rather, it is free to choose the approach that provides optimized circuit operation.

In designing a circuit, the automated data processor constructs an admittance matrix and the first derivative thereof with respect to the design variables. The admittance matrix and the first derivative are then used to generate a continuous analytical objective function in accordance with the performance characteristic options selected by the user. The first derivative of the objective function is also generated by the processor. Both the objective function and its first derivative are called by an optimizer routine that modifies at least one of the design variables in a direction and by an amount dependent on the objective function and the objective function derivative. Using the modified design values, a selected circuit topology is then reconsidered to determine if additional improvement is possible. An optimized circuit design is achieved by considering the selected circuit topology and the objective function over a range of frequencies.

To optimize a particular circuit topology to achieve a given performance characteristic, the objective function includes terms for the sum of a power function of the difference between a specified performance characteristic and a calculated performance over a range of frequencies. Where applicable, the objective function includes a term to minimize the difference between an actual performance characteristic and a specified performance characteristic beyond preset limits. Further, the objective function may include terms such that the final circuit design will operate at a given frequency either at, below, or above a specified value of a performance characteristic.

Further, in accordance with the process of the present invention, an automatic data processing machine will optimize the operation of a circuit that includes elements that may be switched into and out of the circuit, that is, a circuit that has several operating modes. Again, the data processor considers the entire circuit as a whole. An admittance matrix and the first derivative thereof with respect to the design variables is constructed for each of the several operating modes. A composite objective function is then generated using the admittance matrix and the first derivative for each of the several operating modes. The composite objective function is then called by the optimizer routine that modifies design variables throughout the several operating modes in a direction and by an amount dependent on the composite objective function and the objective function derivative. As explained, the design values modified in the optimizer routine are then used to determine if additional improvement is possible.

The improvements in circuit operation achieved by the method of the present invention is primarily the result of generating a continuous analytical objective function that has a continuous first derivative. This continuous analytical objective function and its continuous first derivative are generated in a routine that employs the admittance matrix of a selected circuit topology and the first derivative of the matrix. Both the objective function and the first derivative are inputs to an optimizer routine. In the optimizer routine the original element values are adjusted in a manner to improve a numerical representation of the objective function. The optimizer routine may be repeated several times, each time using a new value of the objective function and its first derivative (generated from design variables modified in the previous optimizer pass) to further adjust the circuit element values. This readjustment continues for a preselected number of passes at a preselected number of frequency points to produce the best possible circuit design.

A more complete understanding of the invention and its advantages will be apparent from the specification and claims, and from the accompanying drawings illustrative of the invention.

Referring to the drawings:

FIG. 1 is a block diagram of a total automated data processing machine for generating circuit masks representing an optimized circuit design;

FIG. 2 is a basic flow chart of the optimization operation in a circuit design sequence;

FIG. 3 is a logic flow chart illustrating the method of optimizing and analyzing a circuit design;

FIG. 4 is a schematic of a single-mode representative circuit topology that may be optimized by the method of the present invention; and

FIG. 5 is a schematic of a multimode circuit which also may be optimized by the method of the present invention.

Referring to FIG. 1, there is shown a block diagram of a system for computerized design of linear time invariant electronic circuits consisting of resistors, capacitors, inductors, transmission lines, controlled current and voltage sources, circulators, transformers, and three terminal devices characterized by s or y parameters as a function of frequency and one other variable. The input to the system is a circuit topology and circuit performance criteria. Outputs include optimized values for the parameters of the circuit topology along with

performance characteristics of the optimized circuit, and for microstrip circuitry a photomask for producing an optimized circuit.

Coded data of a selected circuit topology may be inputted to an analysis and optimization routine 10 from a memory storage 12 or punched cards 14. Catalog circuits in the memory storage 12 may be retrieved by an identifying code number. These circuits have been optimized for particular objectives, and can be reoptimized for different objectives or with different active device characterization data. Also, it is possible to modify the topology of the cataloged circuits by adding, changing, or deleting elements. Use of the catalog of circuits simplifies setting up the analysis and optimization routine. Where the desired circuit topology is not one of the cataloged circuits, however, the desired circuit may be coded onto the punched cards 14 as an input to the analysis and optimization routine 10.

In addition to a circuit topology and initial circuit element values, the analysis and optimization routine 10 may receive input information on active circuit components in the form of s or y parameters as a function of frequency and some second variable. Again, the device data may be inputted to the analysis and optimization routine 10 from a device memory storage 16 or device punched cards 18. In addition to device data and circuit topology and parameters, the analysis and optimization routine 10 must receive input data in the form of a design goal for the selected circuit. This design goal is an expression (objective function) which is in some way proportional to the "goodness" of the circuit. The objective function may be selected from a number of available options or prepared to satisfy a particular design goal. In either case, the objective function is inputted into the analysis and optimization routine 10 by means of punched cards 20.

After receiving input information on a desired circuit topology and initial values of all circuit elements, and at least one desired performance characteristic, the analysis and optimization routine 10 produces an output giving optimized values for the circuit elements and a circuit analysis, when requested, which is an input to a microstrip layout routine 22. For microstrip circuits, the values for the circuit elements in the output from the analysis and optimization routine 10 will include the physical length and width of the transmission lines. As an alternative, the output from the optimization routine 10 may be readout and printed on a layout parameter list 24. The information on the list 24 may then be used to punch layout data cards 26. Layout data cards may then be used at some later time to provide coded information to the microstrip layout routine 22. There usually will be one data card 26 per circuit element giving the line length and width, the nodal connection, and approximate coordinates of the end points. In addition to the optimization routine 10 and the layout cards 26, the microstrip layout routine 22 may also receive layout data from a memory drum 28. The memory drum 28 and/or the data cards 26 supplies data on the approximate configuration of the microstrip lines.

The output of the layout routine 22 consists of coordinates of the end points and bend points of all elements of the circuit topology optimized by the analysis and optimization 10. A mask 30 will be generated outlining the desired circuit topology with the optimized circuit values. This mask will be generated by means of a plotter 32, for example, a Gerber plotter, connected to the output of the layout routine 22. The layout 22 will also produce a listing of data cards 34 which may be punched for future mask generation.

Referring to FIG. 2, there is shown a basic block diagram of a logic flow chart for the optimization routine 10 for a circuit design. The program for optimization of a circuit design includes subroutines that perform nodal analysis on a specified circuit topology coupled with a subroutine of conjugate gradient optimization. After the selected circuit topology, with nodal points identified, and all circuit element values, has been coded and stored, an objective function is generated from the selected objective function options (block 36). In ad-

dition to an objective function, the first derivative or gradient thereof with respect to design variables is generated and stored. Optimization of a circuit design begins by questioning whether the selected circuit is to be optimized or channeled to an output circuit 38. If the inquiry, identified by the diamond 40, produces an answer that the selected circuit requires further optimization, then the objective function and its gradient are called by an optimizer subroutine to optimize the objective function by adjusting circuit design variables, indicated by the block 42. The selected circuit topology with the adjusted design variables is then analyzed and a new objective function and a first derivative are generated by the subroutine of block 36. The analysis compares a synthesized performance characteristic with a desired performance characteristic, as represented by the objective function.

If the desired performance characteristic has not been achieved in the last optimization and adjusting pass, a new objective function and its gradient are generated and stored based on the adjusted values of the circuit elements. Again, an inquiry is made as to whether another optimization pass will be made. If additional optimization of the circuit is desired, another pass will be made through the optimizer subroutine 42. Another adjustment of the circuit elements takes place, and another evaluation of the objective function and its gradient is made. This cycle continues until either the desired performance characteristics have been obtained, or the number of passes reaches a preset number. In either case, the selected circuit topology and final circuit element values are coupled to the output circuit 38.

Referring to FIG. 3, there is shown a logic flow chart for the optimization routine 10 including the generation of objective functions and a circuit admittance matrix. To run the optimization routine, a selected circuit topology together with initial circuit element values must be coded and made available. The suggested procedure for coding a selected topology is to sketch the circuit and label all the nodes and place identifiers on all the elements. It is not necessary that the node numbers be in any special sequence. The input and output nodes may be specified; however, if no value is specified for the input node the lowest numbered node will assume to be the input. If no value is specified for the output node, the highest numbered node is assumed to be the output. The datum node (a reference node) is normally assumed to be the zero node. The source admittance is always connected between the input node and the reference node and the load admittance is always connected between the output node and the reference node. Source and load admittances and black box (s) or (y) parameters are inputted to the optimizer routine as a sequence of measured data points. In order to interpolate between the data points, a polynomial of specified degree is fit through these points such that the mean-squared error is minimized. Referring to the following equation:

$$A_0 + A_1 F + A_2 F^2 + \dots + A_N F^N = Y(F) \quad (1)$$

the coefficients $A_0, A_1, \dots, A_N$, are determined by means of the measured data points and coded and stored in the memory 46 along with the circuit topology and circuit element values. For any frequency F , the admittance $Y(F)$ for the source, load or black box may be calculated using the above equation. As an alternative, the exact data for the source and/or load admittance at each frequency to be used in the optimization routine may be supplied and coded for storage in the memory 46. The optimization routine will then use the exact values of the source and/or load admittance. For a black box, it is possible to specify the s or y parameters as a function of some second variable, such as collector current in the case of a transistor, in addition to the frequency.

In addition to a selected circuit topology and device characterization, the optimization routine must also be supplied with at least one selection from a plurality of objective function options. The objective function may be written as follows:

$$\text{OBJECTIVE FUNCTION} = \sum_k (W_1 g_1 (\text{Gain}) + W_2 g_2 (\text{Noise Figure}) + \text{etc.}) \quad (2)$$

Where the constants W are weighting factors which allow emphasis to be placed on one or the other of the quantities to be optimized, and the g terms are arbitrary functions of the quantity in parenthesis. The term f_k indicates that the objective function is formed by summing the expression at (2) over several frequencies in the range of interest. Various expressions available for use in generating an objective function are listed in table I.

When an engineer is designing an electrical circuit, he has a set of well-defined circuit characteristics and specifications which he desires the completed circuit to achieve. As previously mentioned, the design engineer selects options that outline such specified circuit performance characteristics. For example, his goal might be to maximize the gain over the frequency band for his circuit, subject to the condition that the variation in the gain over that frequency band be small. The design engineer would then examine the list of available optimization options in table I to determine if an option is included which corresponds to his design goals. In this case, he would find that a combination of options 2 and 3 correspond to these exemplary design goals. When more than one option is chosen by the design engineer, as here, he must also assign a weighing factor to each option which signifies the relative importance of one chosen option with respect to another chosen option. That is, the design engineer might determine that the goals of the circuit which he is designing require optimization according to options 4 and 5 of table I. By optimizing according to option 4, however, the characteristics according to option might become less optimized; while, by optimizing 5, 4 might become less optimized, these being unavoidable characteristics of the physical laws of electricity. The design engineer must then assign relative weights to each of the objective function requirements as to the relative importance of options 4 and 5 are equal, an equal weight would be assigned to each. On the other hand, if the design engineer's circuit requirements are such that, although optimization according to options 4 and 5 are desired, the optimization according to the equation of option 5 is more important than that of option 4, a higher weighing factor will be given to option 5. The relative importance given to each by the processing machine is determined by the actual weighing factor assigned to each which is totally in the control of the design engineer to exactly meet his circuit goals. This same procedure is used for any other set of design specifications which the engineer might have for a circuit he is designing. A more detailed explanation of the optimization of characteristics for a specific circuit is later described in detail with respect to FIG. 4.

In addition to the objective function options, the memory 46 must also receive coded data on the number of frequency points at which the circuit topology will be evaluated to meet the selected performance characteristic. Further, the memory 46 must receive instructions as to the maximum number of passes through the optimizer routine for evaluating the selected circuit.

With all the above instructions coded into the system, the optimization routine first inquires as to whether or not optimization of the selected topology is desired. If optimization of the selected circuit topology is to be bypassed, the number of passes through the optimizer will be zero ($N=0$) as indicated by the block 48. In this case, the selected circuit topology data and the values of the circuit elements will bypass the optimization routine, and instead be transferred to a circuit analysis routine to be described.

If N (the number of optimizer passes) is greater than zero, however, then the selected circuit topology will be optimized. To optimize and adjust a selected circuit, the routine generates an admittance matrix (Y -matrix) from the selected circuit topology as identified by the block 50. By using nodal analysis, a full $n \times n$ admittance matrix for the selective circuit may be formed in most cases. To reduce the order of an admittance matrix, series or cascaded elements are reduced to a single 2×2 matrix which is then added to the system Y -matrix. 75

TABLE I--OBJECTIVE FUNCTION OPTIONS

1.....	$\sum f_k - (GN(f_k) - GREF(f_k))^2$	The calculated gain, GN, at the sampled frequencies, f_k , is least-squared-error fit to a reference gain, GREF, fixed by the user. The gain is fit to GFLOAT, a floating flat reference gain, in the sense of least-squared-error. This floating reference gain (listed as the next option) is a variable and must be given a starting value and an upper and lower limit.
2.....	$\sum f_k - (GN(f_k) - GFLOAT)^2$	GFLOAT is the amplitude of the floating reference gain which is fit in option 2. GFLOAT is maximized and must be used in conjunction with option 2. Options 2 and 3 are listed separately to allow differential weighting of the "smoothness" and "overall flat-gain" objectives. GFLOAT-MAX is the maximum or upper limit of GFLOAT.
3.....	$\sum f_k (GFLOAT - GFLOATMAX)$	
4.....	$\sum f_k - (NF(f_k) - SPECNF)^2$	The area under the noise figure squared vs. frequency curve is minimized. The calculated noise figure, NF, at the sampled frequencies, f_k , is least-squared-error fit to a reference noise figure, SPECNF, fixed by the user.
5.....	$\sum f_k - (MMS(f_k))^2$	The area under the source mismatch squared vs. frequency curve is minimized. The calculated source mismatch, MMS, at the sampled frequencies, f_k , is least-squared-error fit to zero source mismatch.
6.....	$\sum f_k - (MML(f_k))^2$	The area under the load mismatch squared vs. frequency curve is minimized. The calculated load mismatch, MML, at the sampled frequencies, f_k , is least-squared-error fit to zero load mismatch.
7.....	$\sum f_k \sum \Delta f_e - (GN(f_k + \Delta f_e) - GREF(\Delta f_e))^2$	The band pass of a filter is fit to specified shape over the frequency band Δf_e at several center frequencies f_k . The calculated gain, GN, at several sampled frequencies, $f_k + \Delta f_e$, where the f_k are center frequencies and the Δf_e define a distribution of sampling about the f_k , is least-squared-error fit to a reference gain, GREF, specified by the user as a function of Δf_e .
8.....	$\sum f_k - ((REAL(Z(IIN, IIN)) - ZSPEC11))^2 + (AIMAG(Z(IIN, IIN)) - ZSPEC11))^2 + ((REAL(Z(IOUT, IOUT)) - ZSPEC22))^2 + (AIMAG(Z(IOUT, IOUT)) - ZSPEC22))^2$	The Z matrix of circuit is matched to a specified Z matrix. The Z matrix, impedance matrix, of circuit at sampled frequencies, f_k , is least-squared-error fit to an impedance matrix ZSPEC11 ZSPEC12 ZSPEC21 ZSPEC22 fixed by the user. IIN is the node number assigned to the input node by the user and IOUT is the node number assigned to output node by the user. AIMAG is program shorthand for the imaginary word.
9.....	$\sum f_k - (PH(f_k) - PHREF(f_k) - PFLOAT - SFLOAT * f_k)^2$	The phase is fit to a floating reference phase. The calculated phase, PH, at sampled frequencies, f_k , is least-squared-error fit to a reference phase. The reference phase is the sum of PHREF, which is specified by the user as a function of f_k , and PFLOAT + SFLOAT * f_k where PFLOAT and SFLOAT are parameters of the optimization which may vary between limits set by the user.

This procedure is known as ABCD reduction. To generate the admittance or Y-matrix, the system goes through the selected circuit, element by element, cumulatively adding the contributions of each element to the matrix. In many cases the admittance matrix for a circuit will be sparse, i.e., many of the entries are zero.

In addition to the admittance matrix, the first derivative of the matrix is formed by the routine of block 50. Further, as indicated by the block 52, the system has a special matrix inverting routine which takes advantage of the sparsity of the Y-matrix to produce an inverted Y-matrix or Z-matrix which can be used to readily obtain most characteristics of a selected circuit.

Assuming that at least one performance characteristic has been selected such that J (the number of chosen objective function options) is not zero, then the routine of block 54 will generate and store an objective function and the first derivative thereof with respect to the design variables. From the objective function and its first derivative, a numerical value thereof will be generated and stored for circuit optimization. The objective function is generated from the options selected from table I. Using the value of the objective function and the value of the objective function gradient, an optimizer subroutine of block 56 adjusts circuit design variables in a manner dependent upon the objective function and its gradient.

The optimizer subroutine (block 56) may be thought of as trying to find the highest point in an m -dimensional space, where m is the number of variables, and the height is the value of the objective function. By placing constraints on the variables, i.e., the circuit elements, boundaries are formed in this m -dimensional space. The optimizer routine of block 56 uses the conjugate gradient of the objective function and the value of the gradient of the objective function in finding the "highest" point, that is, a circuit that operates as near as possible to the selected performance characteristics.

First the objective function and its gradient are evaluated at some feasible starting point in the m -dimensional space. Having initially evaluated the objective function gradient, the next step is to optimize by adjusting a circuit element to produce a value for the objective function in the steepest uphill direction allowed by the constraints. The objective function and its gradients are again evaluated at the new point. If the value of the objective function has not been increased, the subroutine cuts back on its step size and reevaluates the objective function. This interpolation of the step size continues until the objective function is increased or a limit on the number of cut-backs allowed is reached.

The matrix of conjugate directions is then updated by using the newly found better value for the objective function and its gradient. All future steps are taken in the same direct manner as the first step except that instead of the true gradient, the conjugate gradient is used to determine the direction of the step. The conjugate gradient is derived from the true gradient by multiplying by the matrix of conjugate directions. For a more detailed explanation of the optimizer, reference is made to the Ph. D. thesis of Donald Goldfarb, entitled "A Conjugate Gradient Method for Nonlinear Programming," Ph. D. thesis, Princeton University, 1967. The optimization routine continues until it no longer improves the objective function, or the maximum number of passes through the optimization routine has been reached or the allowed time expires. Although a conjugate gradient optimizer has been described, other optimizer routines may be used with the objective function options of table I for optimization of a circuit design. For example, the optimizer routine of R. Fletcher and M. J. D. Powell, entitled "A Rapidly Convergent Descent Method for Minimization," *British Computer Journal*, Volume 6 (1963), pages 163 through 168 may be used. Another example of an acceptable optimizer routine mentioned in the Fletcher and Powell article is a direct search optimization routine commonly identified as the "Spider" routine. The latter routine is described in detail in a paper by R. Hooke and T. A. Jeevee, entitled "Direct Search Solution of Numerical and Statistical Problems," Jour-

nal of the Association for Computing Machinery, Vol. 8, No. 2, pages 212-229, Apr. 1961.

After each pass through the optimizer routine of block 56, the objective function and its conjugate gradient are evaluated in terms of the Z-matrix of the select circuit topology in accordance with reselected options. The various evaluation options available are listed in table II with the equations used as follows (the various terms used in the following equations and the objective function options given in table I are listed in table II):

Power Gain—NCVP=1, phase is same as for NCVP=4

$$G=10 \log (\text{CABS}(Z(\text{IOUT}, \text{IIN}))^{**2} * \text{ABS}(\text{REAL}(\text{YL}) * \text{REAL}(\text{YS})^{*4})) = 10 \log \frac{P_{\text{out}}}{P_{\text{available}}}$$

Current Gain and Phase—NCVP=2

$$A = -Z(\text{IOUT}, \text{IIN}) / Z(\text{IOUT}, \text{IOUT}) = \frac{I_{\text{out}}}{I_{\text{in}}}$$

$$G=20 \log (\text{CABS}(A))$$

Voltage Gain and Phase NCVP=3

$$A = Z(\text{IOUT}, \text{IIN}) / Z(\text{IIN}, \text{IIN}) = \frac{V_{\text{out}}}{V_{\text{in}}}$$

$$G=20 \log (\text{CABS}(A))$$

Source Voltage Gain and Phase NCVP=4

$$A = Z(\text{IOUT}, \text{INN}) * \text{YS} = \frac{V_{\text{out}}}{V_{\text{source}}}$$

$$G=20 \log \text{CABS}(A)$$

$$P=\text{ARG}(A)$$

Phase deviation PD, is calculated from the phase, P; frequency, FR; PFLOAT; and SFLOAT as follows:

$$\text{PD} = P - \text{PFLOAT} - \text{SFLOAT} * \text{FR}$$

Mismatch

Source—Set A=YS and L=IIN

Load—Set A=YL and L=IOUT

$$\text{MM} = 10 \log (\text{ABS}(4 * \text{REAL}(A) * \text{REAL}(1 / Z(L, L) - A)) * \text{CABS}(Z(L, L))^{**2})$$

$$= 10 \log \left(\frac{4 * \text{VSWR}}{(\text{VSWR} + 1)^2} \right)$$

Input Admittance

$$\text{YIN} = 1000 * (1 / Z(\text{IIN}, \text{IIN}) - \text{YS})$$

Output Admittance

$$\text{YOUT} = 1000 * (1 / Z(\text{IOUT}, \text{IOUT}) - \text{YL})$$

Input Impedance

$$\text{ZIN} = 1 / (1 / Z(\text{IIN}, \text{IIN}) - \text{YS})$$

Stability Factor

For the stability factor option, Y2B2(1,1) and Y2B2(2,2) have the source and load subtracted out i.e.

$$\text{Y2B2}(1,1) = \text{Y2B2}(1,1) - \text{YS}$$

$$\text{Y2B2}(2,2) = \text{Y2B2}(2,2) - \text{YL}$$

For loaded stability factor they are not modified

$$\text{SF} = (2 * \text{REAL}(\text{Y2B2}(1,1)) * \text{REAL}(\text{Y2B2}(2,2)) - \text{REAL}(\text{Y2B2}(1,2) * \text{Y2B2}(2,1)) / \text{CABS}(\text{Y2B2}(2,1) * \text{Y2B2}(1,2)))$$

Input VSWR

$$\text{DUM} = \text{CABS}(2 * \text{REAL}(\text{YS}) * Z(\text{INN}, \text{INN}) - 1.0)$$

$$\text{VSWR} = (1 + \text{DUM}) / (1 - \text{DUM})$$

Output VSWR—substitute YL for YS and IOUT for IIN in input VSWR.

Reflection loss and phase angle

$$\text{RHO} = 2 * \text{YS} * (\text{INN}, \text{INN}) - 1.$$

$$\text{ANGLE} = \text{ARG}(\text{RHO})$$

$$\text{LOSS} = 20 \log (\text{CABS}(\text{RHO}))$$

Noise figure and Noise Parameters

$$\text{NF} = \text{NF}_{\text{opt}}$$

$$+ \frac{\text{RN}}{\text{G}_{\text{source}}} \left[(\text{G}_{\text{source}} - \text{G}_{\text{opt}})^2 + (\text{B}_{\text{source}} - \text{B}_{\text{opt}})^2 \right]$$

TABLE II

Definition of terms

Z-MATRIX—A matrix Z(I,J), where I and J are node numbers, defined by the equation $V=ZI$ where V is the vector of node voltages and I is the vector of node currents. Normally called the impedance matrix.

Table II—Continued

Y-MATRIX—A matrix $Y(I,J)$ where I and J are node numbers. It is defined by the equation $I=YV$ where I and V are as above. Normally called the admittance matrix.		
ZS—The source impedance.		
YS— $1/ZS$.		
ZL—The load impedance.		
YL— $1/ZL$.		
IIN—The number of the input node.		
IOUT—The number of the output node.		
Modes—Various states which a circuit may alternate between. For example in a phase shifter, two modes might be the 0° phase shift mode and the 22.5° phase shift mode.		
Y2B2(I,J)—A two by two matrix of admittances formed by taking the inverse of the matrix		
$Z(IIN,IIN)$ $Z(IIN,IOUT)$		
$Z(IOUT,IIN)$ $Z(IOUT,IOUT)$		
ABS—A function which means take the absolute value of the quantity in parenthesis.		
CABS—Same as ABS except the quantity in parenthesis is complex.		
CONJG—A function which means take the conjugate of the quantity in parenthesis.		
ARG—The polar angle of the complex quantity in parenthesis.		
GFLOAT—GFLOAT is a variable which provides a floating reference for the gain of a circuit when optimizing. This allows one to do such things as maximize the flat gain of an amplifier over some frequency band.		
PFLOAT—PFLOAT is a variable which provides a floating reference for the phase of a circuit when optimizing.		
SFLOAT—SFLOAT is a variable which provides a floating reference for the slope of the phase of a circuit when optimizing. Used in conjunction with PFLOAT and a specified phase curve, it allows one to fit the phase deviation from linear to a specified curve.		
KIND NUMBER—The type of an element (e.g. resistor, capacitor, etc.) is specified in the Program Listing by a KIND number. (See page 22.)		
KIND numbers		
Variable Type:	KIND	
GFLOAT-----	-1	45
PFLOAT-----	-2	
SFLOAT-----	-3	
Element Type:		
Black box (up to ten kinds of black boxes allowed)-----	1-10	50
Resistor-----	11	
Capacitor-----	12	
Inductor-----	13	
Series transmission line—BL-----	15	
Shorted transmission line—BL-----	16	
Open transmission line—BL-----	17	55
Series lossy transmission line—BL-----	18	
Shorted lossy transmission line—BL-----	19	
Open lossy transmission line—BL-----	20	
Open radial line—Outer radius-----	23	
Parameter Type:		
Characteristic admittance-----	14	60
Attenuation for lossy line-----	40	
Guided to freespace wavelength-----	41	
Inner radius for radial line-----	42	
Angle for radial line-----	43	
Thickness of substrate material-----	44	65
Dielectric constant of substrate-----	45	

¹ If an element is in a series branch, add 200 to its KIND number.

Of particular significance to the improvement of a circuit design in accordance with the present invention is the various options available for generating the objective function. These objective function options result in the generation and storage of a value which depends on the sum of a power function of the difference between a specified performance characteristic

and a calculated performance characteristic beyond a specified tolerance. Further, a circuit may be adjusted at a particular frequency such that the difference between a specified performance and a calculated performance is zero.

- 5 In addition, where applicable, the calculated performance can be made to be greater than or less than a specified performance characteristic at a given frequency. Consider options 1 and 2 listed in table I. These two options are the same with the exception that for the second option GREF and
- 10 NEQGRF (an index) are always zero. Let:

$$A(f_k) = GN(f_k) - GREF(f_k) - GFLOAT$$

$$A1(f_k) = ABS(A(f_k)) - GTOL$$

$$A2(f_k) = 0 \text{ if } NEQGRF(f_k) = 1 \text{ and } A(f_k) > 0$$

$$A2(f_k) = 0 \text{ if } NEQGRF(f_k) = -1 \text{ and } A(f_k) < 0$$

$$A2(f_k) = 0 \text{ if } A1(f_k) < 0$$

$$A2(f_k) = A1(f_k) \text{ otherwise;}$$

then the contribution of option 1 or option 2 to the objective function is:

$$F = \sum_k (A2(f_k))^2.$$

- 20 Thus for the gain options 1 and 2, the calculated performance characteristic may be made to equal, be less than, or greater than the desired performance characteristic. Similarly, for option 7, a calculated attenuation versus frequency curve may be made to fit a specified curve to the degree required.

- 25 Referring again to that portion of the flow chart for the optimization of a selected circuit topology. Input data in the form of a selected circuit topology and objective function options are inputted to the memory 46 along with the number of passes (N) through the optimization routine and the frequency points (K) to be evaluated.

The optimization routine of Donald Goldfarb is as follows:

Let the current point be x^i with a function value of $f(x^i)$ and a gradient $g^i = g(x^i)$. It is assumed that x^i is in R and lies in the (affine) subspace M_q determined by the intersection of q linearly independent hyperplanes. The matrix operator H is therefore H_q^{-1} . The conjugate-gradient algorithm is then:

1. Compute $H_q^{-1}g^i$,
and $\alpha = (N_q^{-1}N_q)^{-1/2} N_q^{-1}g^i$
If $H_q^{-1}g^i = 0$ and $\alpha \leq 0$, then x^i is global maximum.
2. If this is not the case, either $|H_q^{-1}g^i| > \max\{0, \frac{1}{2}\alpha_q b_{qq}^{-1/2}\}$ or $|H_q^{-1}g^i| \leq \frac{1}{2}\alpha_q b_{qq}^{-1/2}$, where $\alpha_q b_{qq}^{-1/2} \geq \alpha b_{ii}^{-1/2}$, $i=1, \dots, q-1$,

If the former holds proceed to step (3). If the latter applies, drop the q^{th} hyperplane from the constraint basis and obtain H_{q-1}^{-1} from

$$H_{q-1}^{-1} = H_q^{-1} + \frac{P_{q-1}n_q n_q' P_{q-1}}{n_q' P_{q-1} n_q}$$

where $P_{q-1} = I - N_{q-1} N_{q-1}' (N_{q-1}' N_{q-1})^{-1} N_{q-1}'$

Let $q=q-1$ and return to the beginning of step (2).

3. Let $s^i = H_q^{-1}g^i$ and compute λ from the following formulas:

$$\lambda_j = -(n_j' x^i - b_j) / n_j' s^i$$

$$(j=q+1, \dots, k)$$

$$\lambda = \min\{\lambda_j > 0\}$$

Set $\hat{x}^{i+1} = x^i + \lambda s^i$ and compute $\hat{g}^{i+1} = g(\hat{x}^{i+1})$.

4. If $s^i, \hat{g}^{i+1} \neq 0$ and $f(\hat{x}^{i+1}) > f(x^i)$ set $\hat{x}^{i+1} = x^{i+1}$ and $g^{i+1} = \hat{g}^{i+1}$ and add to the constraint basis the hyperplane corresponding to the $\min\{\lambda_j\}$ in step (3) above. H_{q+1}^{-1} is computed as follows:

$$H_{q+1}^{-1} = H_q^{-1} - \frac{H_q^{-1} n_j n_j' H_q^{-1}}{n_j' H_q^{-1} n_j}$$

Set $q=q+1$ and $i=i+1$ and return to step (1).

5. If $s^i, \hat{g}^{i+1} < 0$ obtain $\gamma^i, 0 < 65^\circ \leq \lambda^i$, which maximizes $f(x^i + \gamma^i s^i)$. Set $\sigma = \gamma^i s^i$ and $x^{i+1} = x^i + \sigma^i$. Note that the intersection M_q remains unchanged. Calculate $g^{i+1} = g(x^{i+1})$ and set $y = g^{i+1} - g^i$

Finally calculate value of H_q^{i+1} from

$$H_q^{i+1} = H_q^i + A^i + B^i$$

where

$$A^i = -\frac{\sigma^i \sigma^{i'}}{\sigma^{i'} y^i}$$

$$B^i = -\frac{H_q^i y^i y^{i'} H_q^i}{y^{i'} H_q^i y^i}$$

Set $i=i+1$ and return to step (1).

If the initial feasible point is an interior point of the allowable convex region R , H_q^0 is initially chosen to be any positive definite symmetric matrix, usually I the identity matrix. However, if the initial point lies on the boundary B on exactly q linearly independent hyperplanes H_q^0 is set equal to P_q .

In the above algorithm, b_i is the right-hand side of constraint; b_{ii} is the i^{th} diagonal element of $(N_q' N_q)^{-1}$; f is the objective function; g is the vector gradient; n_j is the unit normal to j^{th} constraint; P is the number of constraints; q is the number of constraints in basis; s is the conjugate gradient direction $s = H_q g$; y^i is the difference in gradients between neighboring points $y^i = g^{i+1} - g^i$; x is the solution variable vector; B is the boundary of feasible region; H_q is the non-Euclidean projection operator basic to the conjugate gradient algorithm; I is the identity matrix; M_q is the linear manifold formed by the intersection of q linearly independent hyperplanes; N_q is the $m \times q$ basis matrix whose columns are the q linearly independent unit normals n_i , $i=1, \dots, q$; P_q is the Euclidean projection operator which maps E^m onto M_q , where:

$$P_q = I - N_q (N_q' N_q)^{-1} N_q'$$

R is the feasible region; α is the vector of Lagrange multipliers (shadow prices) $\alpha = (N_q' N_q)^{-1} N_q' g$; γ is bound on the quadratic term in the equation where:

$$f(x) = f(x_0) + (x - x_0)' g(x_0) + (x - x_0)' (G(x')/2) (x - x_0)$$

where $g(x_0)$ is the gradient of $f(x)$ at x_0 ;

γ^i is the scalar denoting distance in direction s^i ; λ is the distance in positive s direction to nearest constraint (defined in the equation where:

$$\lambda_j = -(n_j' x^i - b_j) / n_j' s^i$$

$$(j=q+1, \dots, k)$$

$$\lambda^i = \min\{\lambda_j > 0\}$$

Set $\hat{x}^{i+1} = x^i + \lambda^i s^i$ and compute $\hat{g}^{i+1} = g(\hat{x}^{i+1})$; k is the total number of inequality constraints; σ is the distance along conjugate gradient direction to maximum in that direction $\sigma = \gamma^i s^i$.

Assuming the use of a computer, such as the UNIVAC 1108, to optimize the selected circuit topology, a program listing of instructions to generate an objective function and its gradient for use in an optimization routine, such as that of Donald Goldfarb, is given in table III. The language of the program is FORTRAN 5. Instructions 36 through 40, 42 through 58, 65 through 70, 83 through 84, and 90 through 96 are scaling and timing instructions which are not required to generate an objective function and have not been included in table III. Other instructions which may not be self-explanatory are explained in table IV.

After the computer has carried out instructions 1 to 249, the objection function and its conjugate gradient are called by an optimizer program (block 56) that optimizes and adjusts the circuit topology design variables, as explained. If the circuit has not been optimized to the desired level or the programmed number of passes (N) has not been exceeded, the computer again utilizes instructions 1 through 249 of the program listing to generate a new objective function and conjugate gradient using the adjusted circuit element values. The new objective function and conjugate gradient are again called by the optimizer program to again optimize the circuit topology by adjusting circuit design variables. This sequence continues until the circuit is optimized to the desired level or the number of passes equals the programmed number (N).

TABLE III

Program Listing

```

1      SUBROUTINE OBJTIV
2      C
3      COMMON/SPECS/GTOL, SPECIN, SPECOT
4      COMMON /CONTRL / IFACT(6), KALPHA, ITYPE, ITYPE1, ICARD,
5      ICODE, IBEGIN,
6      1IUSER, IMODI, ILIST, USERN, IERR, IREDO, IOCARD(80), ISECUR
7      REAL NF, LMDALO
8      COMPLEX      Y, Z(50,50), ZCJG, OMEGA, C1, C2, C3, C4, YBRNCH
9      INTEGER*2 NCOD, NELE, NSTR, KIND, NODE1, NODE2, NODE3, NCOR
10     INTEGER*2 NDBR1, NDBR2, INDEX
11     COMMON G(50), X(200), F, N, NSPI
12     COMMON/ERASE/E(6)
13     COMMON/MATRIX/Y(50,50), ZCJG(51), C1(50), C2(50), C3(50),
14     C4(50)
15     COMMON/CALL/DES(200,2), XMIN(200), XMAX(200), TOL(200),
16     NCOD(200),
17     1 NELE(200), NSTR(200)
18     COMMON/GAIN/GN, DGN(50), GL, PZ, DPZ(50)
19     COMMON/GENRAL/OMEGA, DGRGEN(50)
20     COMMON /DDT/SOURCE(2,26), XLOAD(2,26), NSO, NLO, NNT,
21     NAME(20), LOCATE
22     1(11), BBOX(1600)
23     DIMENSION IBBOX(1600)
24     EQUIVALENCE (IBBOX, BBOX)
25     COMMON/TITLE1/TITLE (18)
26     COMMON/NOISE/NF, DNF(51), YBRNCH(50,4), NDBR1(50),
27     NDBR2(50), NBRNCH

```

Table III - Continued

```

23      1 , SPECNF
24      COMMON/REFREQ/FREQ(60),GREF(60),FOMIN,FIN,NFPTS,LO,
      FR,NFRQOT,FREF,
25      1 NEQGRF(60) ,PHASE(60) ,DELT ,PTOL
26      COMMON/INDEX1/NSW1,NSW3,NCVP,NBL,NTOTAL, NODE,MPLLOT,
      NREF,MREORD,
27      1 INDEX(50,2),NTOT,NVAR,NLARGE
28      COMMON/ELEMET/KIND(200),NODE1(200),NODE2(200),
      NODE3(200),NCOR(200)
29      COMMON/OPTION/NOF(10),CONOPF(10)
30      COMMON/SCAFIL/NCCC,NLVAR,NHVAR,NFILT,SCALE(50),XLLL(50),
      CFREQ(60),
31      1 WIDTH(60),NPTS
32      COMMON/TRALIN/ATEN,LMDALO,RINNER,ER,H
33      EQUIVALENCE(Y(1,1),Z(1,1)),(NTT,NNT),(T1,E(1))
34      COMPLEX DELZ
35      EQUIVALENCE(NNODE,NLARGE)
36-40
41      EEE= .4342945
42-58
59      1200 CONTINUE
60      F=0.
61      C
62      DO 1001 I=1,N
63      1001 G(I)=0.
64      C
65-69
70      NGO = 1
71      IF(LO.EQ.3)NGO=2
72      ND=0
73      GO TO(1003,1004),NGO
74      1003 ND=NFRQOT
75      NST=1
76      1004 DO 1002 MODE=1,NFILT
77      GO TO (1005,1006),NGO
78      1006 NST=ND+1
79      ND=ND+NFRQOT
80      1005 DO 1002 K=NST,ND
81      GN=0.
82      C
83
84
85      C
86      CALL YSTART(K,MODE)
87
88      C
89      CALL CMTINV(NODE,NLARGE)
90-96
97      C
98      202 CONTINUE
99      DO 73 L=1,NNODE
100      73 ZCJG(L)=CONJG(Z(NNODE,L))
101      ZCJG(51)=CONJG(Z(1,1))
102      C
103      DO 74JN=1,10
104      C
105      KK=NOF(JN)+1
106      GO TO (1002,301,301,301,4,5,6,7,8,74,74,11),KK
107      301 DUM1 = 0.
108      DO 302 I = 1,NV
109      IDUM = I
110      IF(KIND(I).EQ.-1) GO TO 303

```

Table III - Continued

```

111 302 CONTINUE
112 IDUM = 0
113 GO TO 304
114 303 DUM1 = X(IDUM)
115 304 IF(KK-3) 1,1,3
116 C
117 1 CALL GNCALC(KR)
118 CALL DGNCLC(KR)
119 GO TO (74,101,2),KK
120 101 KREF=K
121 IF(K.GT.60) KREF= 60
122 DUM=GL-GREF(KREF)
123 C ***IF GFLOAT IS SPECIFIED IN THE CIRCUIT ELEMENTS THEN
SUBTRACT IT OUT
124 C FOR A FLOATING GAIN CURVE.
125 222 IF(IDUM) 223,223,224
126 224 IF(KIND(IDUM).EQ.-1) DUM = DUM - DUM1
127 IF(KK.EQ.3) GO TO 110
128 223 CONTINUE
129 IF(NEQGRF(KREF)) 111,110,113
130 111 IF(DUM)74,74,112
131 113 IF(DUM)112,74,74
132 110 AUM=ABS(DUM)-GTOL
133 IF(AUM.LE.0.)GO TO 74
134 DUM=SIGN(AUM,DUM)
135 112 F=F - CONOPF(JN)*DUM*DUM
136 GDUM=20.*DUM*CONOPF(JN)*EEE/GN
137 DO 1010 I=1,N
138 1010 G(I)=G(I)- GDUM*DGN(I)
139 IF(IDUM.GT.0) G(IDUM) = G(IDUM) + CONOPF(JN)*2,*DUM
140 GO TO 74
141 C
142 2 DUM = GL
143 GO TO 222
144 C
145 C MAXIMIZE GFLOAT
146 3 DUM = XMAX(IDUM)
147 IF(NBL.EQ.2) DUM = XL11(IDUM) + SCALE(IDUM)
148 F = F + CONOPF(JN) *(DUM1 -DUM)
149 G(IDUM) = G(IDUM) + CONOPF(JN)
150 GO TO 74
151 C
152 4 CALL NFCLC(MODE)
153 CALL DNFCLC(MODE)
154 IF(SPECNF.GE.0.)GO TO 1014
155 DUM=10.*ALOG10(NF)+SPECNF
156 GO TO 1013
157 1014 DUM=10.*ALOG10(NF)-SPECNF
158 IF(DUM)74,74,1013
159 1013 F=F-CONOPF(JN)*DUM*DUM
160 GDUM=20.*DUM*CONOPF(JN)*EEE/NF
161 DO 1011 I=1,N
162 1011 G(I)=G(I)-GDUM*DNF(I)
163 GO TO 74
164 C
165 5 KINDX=1
166 CALL MISMIN(XMISN,KR,KINDX,1)
167 FACT=10.*ALOG10(ABS(XMISN))-SPECIN
168 IF(FACT)1015,74,74
169 1015 F=F-FACT*FACT*CONOPF(JN)
170 GDUM=20.*FACT*CONOPF(JN)/XMISN*EEE
171 DO 20 I=1,N

```

Table III-Continued

```

172      20 G(I)=G(I)-GDUM      *DGRGEN(I)
173      GO TO 74
174      C
175      6 KINDX=2
176      CALL MISMIN(XMISN,KR,KINDX,1)
177      FACT=10.*ALOG10(ABS(XMISN)) -SPECOT
178      IF(FACT)1016,74,74
179      1016 F=F-FACT*FACT*CONOPF(JN)
180      GDUM=20.*FACT*CONOPF(JN)/XMISN*EEE
181      DO 21 I=1,N
182      21 G(I)=G(I)-GDUM*DGRGEN(I)
183      GO TO 74
184      7 IF(K.NE.MODE) GO TO 74
185      KK = LO
186      DO 1310 NPT=1,NPTS
187      FREQ(K)=CFREQ(K)+WIDTH(NPT)
188      CALL YSTART(K,MODE)
189      CALL CMTINV(NODE,NLARGE)
190      DO 1373 L=1,NNODE
191      1373 ZCJG(L)=CONJG(Z(NNODE,L))
192      ZCJG(51)=CONJG(Z(1,1))
193      CALL GNCALC(K)
194      CALL DGNCLC(K)
195      KREF = NPT
196      IF(KREF.GT.60) KREF= 60
197      DUM=GL-GREF(KREF)
198      IF(NEQGRF(KREF)) 114,117,116
199      114 IF(DUM)1310,1310,115
200      116 IF(DUM)115,1310,1310
201      117 AUM=ABS(DUM)-GTOL
202      IF(AUM.LE.O.)GO TO 74
203      DUM=SIGN(AUM,DUM)
204      115 F=F-DUM*DUM*CONOPF(JN)
205      GDUM=20.*DUM*EEE*CONOPF(JN)/GN
206      DO 1309 I=1,N
207      1309 G(I)=G(I)- GDUM*DGN(I)
208      1310 CONTINUE
209      LO = KK
210      FREQ(K) = CFREQ(K)
211      GO TO 74
212      8 CALL FITYOZ
213      GO TO 74
214
215
216      11 IF(GN.NE.O.)GO TO 711
217      CALL GNCALC(KR)
218      CALL DGNCLC(KR)
219      KREF = K
220      IF(K.GT.60) KREF = 60
221      711 DUM1 = 0.
222      DUM2 = 0
223      DO 712 I = 1,NV
224      IF(KIND(I)+2) 713,714,713
225      714 DUM1 = X(I)*.01747737
226      DPZ(I) = -.01747737
227      713 IF(KIND(I)+3) 712,715,712
228      715 DUM2 = X(I)*.01747737
229      DPZ(I) = -FR*.01747737
230      712 CONTINUE
231      DUM = DUM1 + DUM2 *FR
232      717 IF(DUM.LE.3.14159265) GO TO 716
233      DUM = DUM - 6.2831853

```

Table III—Continued

```

234      GO TO 717.
235      716 IF(DUM.GT.-3.14159265) GO TO 718
236          DUM = DUM + 6.2831853
237          GO TO 716
238      718 DUM = PZ - PHASE(KREF) - DUM
239          IF(ABS(DUM).GT.4.7) DUM = DUM - SIGN(6.283185 ,DUM)
240          AUM=ABS(DUM)-PTOL
241          IF(AUM.LE.O.)GO TO 74
242          DUM=SIGN(AUM,DUM)
243          F=F-CONOPF(JN)*DUM*DUM
244          GDUM=2.*CONOPF(JN)*DUM
245          DO 1012 I=1,N
246      1012 G(I)=G(I)-GDUM*DPZ(I)
247          74 CONTINUE
248          GO TO 1002
249      1002 CONTINUE

```

TABLE IV

Instruction number	Explanation
86-----	This instruction calls for a subroutine to form the Y-matrix of the selected circuit topology at a frequency K and for a given mode.
89-----	This instruction calls for a subroutine to invert the Y-matrix of Instruction 86 and to form the Z-matrix.
117-----	This instruction calls for a subroutine to calculate the gain at a frequency KR.
118-----	This instruction calls for a subroutine to calculate the derivative of the gain calculated at Instruction 117 at a frequency KR.
152-----	This instruction calls for a subroutine to calculate the noise figure for the mode under consideration.
153-----	This instruction calls for a subroutine to calculate the derivative of the noise figure with respect to the element values.
166-----	This instruction calls for a subroutine to calculate the mismatch and the derivative of the mismatch at the source (Kind X-1 indicates source) and a frequency KR.
172-----	This instruction calls for Instruction 166 to be stored in the array DGRGN.
176-----	This instruction calls for a subroutine to calculate the mismatch and the derivative for the load (Kind X-2 indicates load) at a frequency KR.
182-----	The mismatch calculated at Instruction 176 is stored in the array DGRGN.
188-----	This instruction calls for the same subroutine as called at Instruction 86 to be evaluated at additional frequency points.

TABLE IV—Continued

Instruction number	Explanation
25-----	189----- This instruction calls for the same subroutine as called by Instruction 89 to be used in conjunction with Instruction 188.
30-----	193----- This instruction calls for the same subroutine as called at Instruction 117.
194-----	This instruction calls for the same subroutine as called by Instruction 118.
212-----	This instruction calls for a subroutine that generates a deviation of the Z-matrix of a circuit from some specified Z-matrix. This subroutine is given in Table V.
217-----	This instruction calls in a subroutine to calculate the gain and phase of the selected circuit topology at a frequency KR.
35-----	218----- This instruction calls for a subroutine to calculate the derivative of the gain and phase for a selected circuit topology at a frequency KR.

NOTE 1.—The margin numbers at Instruction 117, 142, 146, 152, 165, 175, 184, 212, and 216 refer to the objective function options listed in Table I.

NOTE 2.—Any of the subroutines called by the program for one objectives function will not be called for another objective function since the calculation is already stored.

NOTE 3.—Instructions 1-35 establish common areas, equivalencies and type statements.

TABLE V

Program Listing

```

1      SUBROUTINE FITYOZ
2      COMPLEX DELZ,DZ(2,2),Y(2,2),DY(2,2),DUM(2,2)
3      INTEGER*2KIND,NODE1,NODE2,NODE3,NCOR
4      COMPLEX ZCJG,C1,C2,C3,C4
5      COMMON /MATRIX/ Z(50,50),ZCJG(51),C1(50),C2(50),
6      C3(50),C4(50)
7      COMMON/ELEMET/KIND(200),NODE1(200),NODE2(200),NODE3(200),
8      NCOR(200)
9      COMMON G(50),X(200),F,M
10     COMMON/MATCH/NAMEY(2),NZOY,WEIGHT(8),ZTOL,LZOY,SECPZY
11     COMMON/SUBCOM/AB(4),KINDS,K,WF,Y1,Y2,Y3,Y4
12     COMPLEXZ,Y1,Y2,Y3,Y4,Z1,Z2,Z3,Z4,ZS1,ZS2,ZS3,ZS4
13     COMMON/INDEX1/NSW1,NSW2,NCVP,NBL,NTOTAL,NNODE,MPL0T,
14     NREF,MREORD,
15     1 INDEX(50,2),NTOT,NVAR,NLARGE,NGO
16     INTEGER*2 INDEX
17     EQUIVALENCE (DY(1,1),DZ(1,1)),(Y(1,1),Z1),(Y(1,2),Z2),
18     (Y(2,1),Z3),
19     1(Y(2,2),Z4)

```

Table V—Continued

```

16      KINDS = LZ0Y
17      SAVE= X(200)
18      X(200) = SECPZY
19      K = 200
20      CALL GENFIT
21      C GENFIT FINDS Y OF BLACK BOX(KINDS=LZ0Y)
22      X(200) = SAVE
23      C SAVE THE FOLLOWING 4 VALUES WHILE FORMING 2BY2 Y,Z
24      ZS1 = Z(1,1)
25      ZS2 = Z(1,2)
26      ZS3 = Z(2,1)
27      ZS4 = Z(2,2)
28      GO TO (1,2),NZ0Y
29      C 1 FITY, 2FITZ
30      1 Z(1,2) = Z(1,NNODE)
31      Z(2,1) = Z(NNODE,1)
32      Z(2,2) = Z(NNODE,NNODE)
33      C THE Z MATRIX IS NOW CONVERTED TO THE Y MATRIX FOR
      THE CIRCUIT
34      CALL CMTINV(2,2)
35      C THE FOLLOWING 4 VALUES ARE Y'S
36      Z1 = Z(1,1)
37      Z2 = Z(1,2)
38      Z3 = Z(2,1)
39      Z4 = Z(2,2)
40      GO TO 3
41      2 Z1 = Z(1,1)
42      Z2 = Z(1,NNODE)
43      Z3 = Z(NNODE,1)
44      Z4 = Z(NNODE,NNODE)
45      Z(1,1) = Y1
46      Z(1,2) = Y2
47      Z(2,1) = Y3
48      Z(2,2) = Y4
49      C TURN MATCHING MATRIX TO A Z MATRIX
50      CALL CMTINV(2,2)
51      C THE FOLLOWING 4 VALUES ARE Z'S
52      Y1 = Z(1,1)
53      Y2 = Z(1,2)
54      Y3 = Z(2,1)
55      Y4 = Z(2,2)
56      3 Z(1,1) = ZS1
57      Z(1,2) = ZS2
58      Z(2,1) = ZS3
59      Z(2,2) = ZS4
60      C THE ABOVE 4 STATEMENTS RETURN TO ORIGINAL VALUES
61      C ZS1 STORES THE DIFFERENCE IN Y'S FOR NZ0Y=1
62      C ZS1 STORES THE DIFFERENCE IN Z'S FOR NZ0Y=2
63      ZS1 = Z1 - Y1
64      ZS2 = Z2 - Y2
65      ZS3 = Z3 - Y3
66      ZS4 = Z4 - Y4
67      F = F -(REAL(ZS1)**2*WEIGHT(1)+AIMAG(ZS1)**2*WEIGHT(2)
68      1 + REAL(ZS2)**2*WEIGHT(3) + AIMAG(ZS2)**2*WEIGHT(4)
69      2 + REAL(ZS3)**2*WEIGHT(5) + AIMAG(ZS3)**2*WEIGHT(6)
70      3 + REAL(ZS4)**2*WEIGHT(7) + AIMAG(ZS4)**2*WEIGHT(8))
71      IF(M.EQ.0)RETURN
72      DO 204 I=1,M
73      C THE FOLLOWING DOUBLE DO LOOP CALCULATES THE GRADIENT
      OF Z W.R.T.ELE
74      C MENT I
75      DO 201 M1=1,2

```

Table V - Continued

```

76      DO 201 M2=1,2
77      N1=M1
78      N2=M2
79      IF(N1.GE.2)N1=NNODE
80      IF(N2.GE.2)N2=NNODE
81      L= NODE1(I)
82      J= NODE2(I)
83      JJ=NODE3(I)
84      DELZ = (0.,0.)
85      IF(NREF.GT.0) GO TO 300
86      IF(KIND(I).GT.10) GO TO 212
87      IF(JJ.LE.0) GO TO 128
88      C   BOTH L AND J WILL BE NON ZERO FOR BLACK BOX
89      GO TO 301
90      C
91      300 IF(JJ.EQ.NREF)GO TO 212
92      C   NOTE THAT SERIES LINES IN SUMMED BRANCHS CAN NOT BE
          VARIABLE
93      II=I
94      IF(KIND(I).EQ.14) II=NCOR(I)
95      IF((KIND(II).GT.10).AND.(KIND(II).NE.15).AND.(KIND(II).
          NE.18)) GO
96      1 TO 209
97      IF(L.EQ.NREF) L=0
98      IF(J.EQ.NREF) J=0
99      IF(KIND(I).GT.10) JJ=NREF
100     C
101     301 DELZ=-(C1(I)+C2(I)+C3(I)+C4(I))*Z(N1,JJ)*Z(JJ,N2)
102     IF(L.NE.0)DELZ=DELZ+(C1(I)+C2(I))*Z(N1,L)*Z(JJ,N2)+
          (C1(I)+C3(I))*Z
103     1(N1,JJ)*Z(L,N2)
104     IF(J.NE.0)DELZ=DELZ+(C3(I)+C4(I))*Z(N1,J)*Z(JJ,N2)+
          (C2(I)+C4(I))*Z
105     1(N1,JJ)*Z(J,N2)
106     GO TO 212
107     209 IF(L.EQ.NREF) L = 0
108     210 IF ( J.NE.0) GO TO 211
109     J = NREF
110     C2(I) =-C1(I)
111     C3(I) = C2(I)
112     C4(I) = C1(I)
113     GO TO 212
114     211 IF( J.EQ.NREF) J =0
115     212 IF(L.GT.0) GO TO 126
116     DELZ=-Z(N1,J)*Z(J,N2)*C4(I) +DELZ
117     GO TO 200
118     126 IF(J.GT.0) GO TO 128
119     DELZ= -Z(N1,L)*Z(L,N2)*C1(I) +DELZ
120     GO TO 200
121     128 DELZ=-Z(N1,L)*Z(L,N2)*C1(I)-Z(N1,L)*Z(J,N2)*C2(I)-Z
          (N1,J)*Z(L,N2)*
122     1C3(I) -Z(N1,J)*Z(J,N2)*C4(I) + DELZ
123     200 DZ(M1,M2)=DELZ
124     201 CONTINUE
125     SIGN=-1.
126     GO TO(202,203),NZOY
127     202 CALL MATMUL(Y,DZ,DUM,2,2)
128     CALL MATMUL(DUM,Y,DY,2,2)
129     SIGN=1.
130     203 G(I)=G(I)+2,*(WEIGHT(1)*REAL(ZS1)*REAL(DY(1,1))+
          WEIGHT(2)*AIMAG(ZS
131     11)*AIMAG(DY(1,1))+WEIGHT(3)*REAL(ZS2)*REAL(DY(1,2))+
          WEIGHT(4)*AIMA

```

Table V—Continued

```

132      2G(ZS2)*AIMAG(DY(1,2))+WEIGHT(5)*REAL(ZS3)*REAL(DY
      (2,1))+WEIGHT(6)*
133      3AIMAG(ZS3)*AIMAG(DY(2,1))+WEIGHT(7)*REAL(ZS4)*REAL
      (DY(2,2))+WEIGHT
134      4(8)*AIMAG(ZS4)*AIMAG(DY(2,2))) *SIGN
135      204 CONTINUE
136      RETURN
137      END

```

TABLE VI.—EXPLANATION OF INSTRUCTIONS OF TABLE V

Instruction number	Explanation
20.....	This instruction calls for a subroutine to generate a specified Y-matrix (black box matrix) as explained at Instruction 21.
34.....	This instruction calls for a subroutine to invert the Y-matrix of Instruction 20.
50.....	This instruction calls for the same subroutine as called by Instruction 34.
127.....	This instruction calls for a subroutine to multiply the Y-matrix by the derivative of the Z-matrix and stores the result in array DUM. This instruction is skipped if the routine is calling for a Z-matrix fit.
128.....	This instruction calls for a subroutine that multiplies the Y-matrix by the derivative of the Y-matrix and stores the result in array DUM.
136.....	This instruction returns the computer to Instruction 74 of Table III.

NOTE.—Instructions 30 through 40 and 41 through 55 are alternate sets of instructions. Either one or the other of the sets will be used, but not both.

After a given circuit topology has been optimized in a routine that generates and stores an admittance matrix and its derivatives and forms an objective function and its derivative, coded information representing the circuit topology and the final circuit element values are transferred to an analysis subroutine. In particular, the information is transferred to an analysis information section 58. The circuit is now analyzed at selected frequency points and in accordance with selected analysis options listed in table VII. For each selected frequency, the final optimized circuit and the circuit element values are analyzed (block 60) to determine the final circuit performance. After analysis at the last frequency point (k_{max}) coded information representing the final circuit is transferred to either a calculator-computer plotter 62, a circuit layout routine 64, or readout and stored. At the completion of any of these last three options, a signal is transferred to the memory 46 to begin the optimization routine on another selected circuit topology.

Table VII—Continued

```

INPUT VSWR
OUTPUT VSWR
REFLECTED PHASE ANGLE (degrees)
20 REFLECTION LOSS
INPUT RESISTANCE (ohms)
INPUT REACTANCE (ohms)
LOADED STABILITY FACTOR
USER SUPPLIED OPTION
25 NOISE PARAMETERS
PHASE DEVIATION
S11
S12
S21
30 S22
Y11
Y12
Y21
35 Y22

```

*NCVP=1 for power gain, for phase of source voltage gain=2 for current gain=3 for voltage gain=4 for source to load voltage gain.

Referring to FIG. 4, the circuit illustrated is an example of a typical topology that may be optimized by the method of the present invention. The circuit includes shorted transmission lines 2a, 8a, 12a and 16a. In addition, the circuit includes series transmission lines 4a, 6a, 10a and 14a. Two transistors 17a and 18a complete a two-stage broad band amplifier. Assume that the elements and parameters have the starting values given in table VIII with the characteristic admittances (Y_0) in millimhos and electrical lengths (βL) measured in fractions of a wavelength at 1.5 GHz. The source impedance is fixed at 20.000+J.000 millimhos and the load impedance fixed at 20.000+J.000 millimhos. Elements 17a and 18a are transistors with the following S parameters:

Frequency	S11	S12	S21	S22	NQ(DB)	RN	YOFT
.500.....	.2460 238.0	.0400 68.0	6.3900 95.1	.6410 353.0	3.000	150.000	3.000-2.000
1.000.....	.2530 209.0	.0670 71.2	3.3200 80.7	.6420 353.0	3.300	130.000	5.600-7.600
1.500.....	.2750 185.0	.0890 73.2	2.3500 70.7	.6430 353.0	3.600	75.000	9.600-14.400
2.000.....	.2980 186.0	.1190 79.7	1.5600 61.7	.6770 357.0	4.500	20.000	28.000-24.000
2.500.....	.3060 185.0	.1456 84.3	1.2100 53.1	.6820 357.0	5.000	20.000	40.000-32.000
3.000.....	.3260 166.0	.1680 85.9	1.0100 49.0	.6860 359.0	6.000	20.000	60.000-36.000
3.500.....	.3710 164.0	.1940 83.3	.9290 44.4	.7030 364.0	6.600	20.000	80.000-40.000
4.000.....	.3607 154.0	.1940 84.7	.8130 38.4	.7040 366.0	7.500	20.000	100.000-40.000

TABLE VII. OPTIONS FOR ANALYSIS

GAIN—TYPE DETERMINED BY NCVP*—(db.)
 NOISE FIGURE—(db.)
 MISMATCH AT SOURCE—(db.)
 MISMATCH AT LOAD—(db.)
 INPUT CONDUCTANCE—(mmhos)
 INPUT SUSCEPTANCE (mmhos)
 OUTPUT CONDUCTANCE (mmhos)
 OUTPUT SUSCEPTANCE (mmhos)
 STABILITY FACTOR
 PHASE—TYPE DETERMINED BY NCVP* (degrees)

65 This is fit to a fifth order polynomial in frequency
 Referring to table I, options 1, 3 and 4 have been selected for the objective function, each having an equal weight distribution. This circuit was optimized at the following eight frequencies:

70	f=	0.90 GHz
		1.00
		1.20
		1.40
		1.60
		1.80
		2.00
		2.10

TABLE VIII

Element number	Kind code	Description	Nodes			NCOR	Starting value	Minimum bound	Maximum bound	Tolerance
			(1)	(2)	(3)					
19.....	-1	GFLOAT	0	0	0	0	20.0000	18.0000	25.0000	.000000
2.....	16	LSTB1	1	0	0	1	.287185	.100000-01	.490000	.500000-01
3.....	14	Y SER1	1	2	-0	4	10.0402	10.0000	30.0000	.500000-01
4.....	15	L SER1	1	2	0	3	.795181-01	.100000-01	.240000	.500000-01
5.....	14	Y SER2	3	4	-0	6	10.0000	10.0000	30.0000	.500000-01
6.....	15	L SER2	3	4	0	5	.165011	.100000-01	.240000	.500000-01
7.....	14	Y STB2	4	0	-0	8	10.00000	10.0000	30.0000	.500000-01
8.....	16	LSTB2	4	0	0	7	.211851	.100000-01	.490000	.500000-01
9.....	14	Y SER3	4	5	-0	10	.18.4439	10.0000	30.0000	.500000-01
10.....	15	L SER3	4	5	0	9	.500000-01	.500000-01	.240000	.500000-01
11.....	14	Y STB3	5	0	-0	12	10.0000	10.0000	30.0000	.500000-01
12.....	16	LSTB3	5	0	0	11	.827344-01	.100000-01	.490000	.500000-01
13.....	14	Y SER4	6	7	-0	14	17.6546	10.0000	30.0000	.500000-01
14.....	15	L SER4	6	7	0	13	.978295-01	.100000-01	.240000	.500000-01
15.....	14	Y STB4	7	0	-0	16	15.8530	10.0000	30.0000	.500000-01
16.....	16	LSTB4	7	0	0	15	.714040-01	.100000-01	.490000	.500000-01
1.....	14	Y STB1	1	0	-0	2	22.9142	10.0000	30.0000	.500000-01
17.....	1	TRAN-2	5	6	0	0	.000000	.00000	.00000	.000000
17.....	1	TRAN-1	2	3	0	0	.000000	.00000	.00000	.000000

NOTE.—Unless otherwise specified through NCOR, the following parameter values are used: ATEN=.40000-01; LMDALO=.40000; RINNER=.10000-01; H=.20000-01; ER=9.6000.

TABLE IX

Variable No.	Element No.	Description	Optimized values of X	Range of X	
				Lower	Upper
1.....	19	GFLOAT	18.00000	18.00000	25.00000
2.....	2	LSTB1	.287979	.01000	.49000
3.....	3	Y SER1	10.00000	10.00000	30.00000
4.....	4	L SER1	.03479	.01000	.24000
5.....	5	Y SER2	10.00075	10.00000	30.00000
6.....	6	L SER2	.13969	.01000	.24000
7.....	7	Y STB2	10.00000	10.00000	30.00000
8.....	8	LSTB2	.27803	.01000	.49000
9.....	9	Y SER3	19.21691	10.00000	30.00000
10.....	10	L SER3	.05451	.05000	.24000
11.....	11	Y STB3	10.00000	10.00000	30.00000
12.....	12	LSTB3	.11538	.01000	.49000
13.....	13	Y SER4	10.92270	10.00000	30.00000
14.....	14	L SER4	.13994	.01000	.24000
15.....	15	Y STB4	15.35098	10.00000	30.00000
16.....	16	LSTB4	.08314	.01000	.49000
17.....	1	Y STB1	21.60968	10.00000	30.00000

TABLE X

Frequency	Gain (db)	NF (db)	Phase	VSWR	
				In	Out
.8000.....	21.02	6.811	-112.8	2.275	13.70
.8500.....	20.61	6.291	-129.8	2.318	11.66
.9000.....	20.11	5.830	-145.5	2.329	10.28
.9500.....	19.58	5.439	-160.0	2.340	9.272
1.0000.....	19.06	5.126	-173.4	2.368	8.406
1.050.....	18.56	4.895	173.9	2.418	7.861
1.100.....	18.10	4.747	162.0	2.492	7.315
1.150.....	17.69	4.677	150.6	2.587	6.824
1.200.....	17.35	4.675	139.5	2.698	6.369
1.250.....	17.07	4.727	128.8	2.823	5.937
1.300.....	16.85	4.820	118.2	2.958	5.520
1.350.....	16.69	4.937	107.7	3.097	5.113
1.400.....	16.59	5.066	97.24	3.238	4.712
1.450.....	16.55	5.196	67.61	3.372	4.315
1.500.....	16.56	5.317	75.78	3.492	3.922
1.550.....	16.62	5.424	64.63	3.587	3.533
1.600.....	16.71	5.512	53.06	3.642	3.150
1.650.....	16.83	5.578	40.97	3.639	2.776
1.700.....	16.95	5.621	28.23	3.562	2.416
1.750.....	17.04	5.642	14.77	3.400	2.076
1.800.....	17.09	5.643	.5219	3.155	1.762
1.850.....	17.05	5.625	-14.51	2.851	1.483
1.900.....	16.88	5.593	-30.23	2.524	1.247
1.950.....	16.57	5.551	-46.48	2.219	1.097
2.000.....	16.09	5.504	-63.03	1.978	1.190
2.050.....	15.46	5.460	-79.66	1.834	1.370
2.100.....	14.69	5.426	-96.24	1.808	1.551
2.150.....	13.80	5.416	-112.8	1.899	1.709
2.200.....	12.83	5.446	-129.4	2.093	1.830

TABLE XI.—DIMENSIONS OF TRANSMISSION LINES IN INCHES
[FREF=1.50000]

Element number	Thickness of substrate	Relative dielectric constant	Width of line	Length of line
2.....	.200000-01	9.60000	.232394-01	.831668
4.....	.200000-01	9.60000	.284440-02	.113357
6.....	.200000-01	9.60000	.284522-02	.455188
8.....	.200000-01	9.60000	.284440-02	.905961
10.....	.200000-01	9.60000	.183638-01	.170012
12.....	.200000-01	9.60000	.284440-02	.375970
14.....	.200000-01	9.60000	.393557-02	.453978
16.....	.200000-01	9.60000	.109287-01	.264105

After completion of twenty optimization passes through the routine (N=20), the optimized values of the circuit elements are given in table IX. Using the values of table IX and analyzed for a power gain (NCVP=1), the analysis results are given in table X. In addition to optimization and analysis results, a computer for optimization of the circuit of FIG. 4 also generated a readout giving the dimensions in inches for the transmission lines. These are listed in table XI and comprise the information used by the layout option 64, which may be a Gerber Plotter, to produce a pattern for fabricating the illustrated circuit.

In addition to optimizing a circuit of the type shown in FIG. 4, the method of the present invention may also be used for optimizing a circuit that has several switching states or modes, that is, elements that may be switched into and out of the circuit. Referring to FIG. 5, a circuit is shown having four switching modes or states. An input impedance 88 connects directly to one pole of a double-pole double-throw switch 90 and to the second pole through an inductor 92. Each branch of the circuit shown includes a resistor 94 in series with a capacitor 96. The first four branches are interconnected and connect directly to one pole of a double-pole double-throw switch 98 and to the second pole through an inductor 100. The second four branches are interconnected to an output impedance 104. Again, the circled numbers are the nodes for analysis and admittance matrix generation for the optimization routine.

Optimization of a multimode circuit of a type illustrated in FIG. 5 is similar to the optimization of a single-mode circuit. In addition to the previously described input information to the memory 46, a multimode circuit requires information identifying the number of modes. With reference to the program listing of table III, the data processor completes instructions 1 through 249 for mode one with the matrix generated from the circuit elements in the circuit in the first mode. This same routine is repeated for mode two using the circuit elements of the second mode to generate the admittance matrix and its derivative with respect to the design variables. For the circuit of FIG. 5, the routine set out in table III is repeated four times. Upon completion of the fourth mode, a composite value of the objective function and a composite value of its conjugate gradient are generated and stored for use by the optimizer routine of block 56.

In the optimizer routine, all the circuit elements in the various modes may be adjusted in a manner and a direction dependent upon the objective function and its conjugate gradient. If additional passes are to be made through the optimization routine, the program listing of table III is again used to evaluate the circuit in each of its four modes. In the logic flow chart of FIG. 3, the multimode instructions appear immediately below the optimizer routine block 56.

Comparing the optimization routine of a multimode circuit and a single mode circuit, the significant difference is in the

generation of the objective function and its conjugate gradient. The same options as given in table I may be used for generating the objective function. In a single mode circuit, the subroutine given in table III is completed once for each optimization pass. For a multimode circuit, the subroutine of table III is repeated once for each mode for each optimization pass.

Having described the invention in accordance with applicable United States statutes, we claim:

1. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

- a. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial element values,
- b. generating and storing a value of said continuous objective function which depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits, and
- c. modifying at least one of the design variables in a direction and by an amount dependent on said objective function to improve the generated and stored value thereof.

2. The method of designing a circuit by an automated data processing machine as set forth in claim 1 wherein the generated and stored value of said continuous objective function depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits and a greater than, less than or equal to difference between the specified performance characteristic and the synthesized performance characteristic.

3. The method of designing a circuit by an automated data processing machine as set forth in claim 1 including repeating steps (a), (b) and (c) for a preselected number of passes utilizing the design variables as modified in the previous pass to generate and store a new objective function and to further improve the generated and stored value of the new objective function.

4. The method of designing a circuit by an automated data processing machine as set forth in claim 3 including the step of analyzing the selected circuit topology at the final values of the circuit elements.

5. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

- a. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial element values,
- b. generating and storing a value of said continuous objective function which depends on the sum, over a range of frequencies, of a power function wherein a synthesized performance characteristic may be less than, greater than, or equal to a specified performance characteristic, and
- c. modifying at least one of the design variables in a direction and by an amount dependent on said objective function to improve the generated and stored value thereof.

6. The method of designing a circuit by an automated data processing machine as set forth in claim 5 including repeating steps (a), (b) and (c) for a preselected number of passes utilizing the design variables as modified in the previous pass to generate and store a new objective function and to further improve the generated and stored value of the new objective function.

7. The method of designing a circuit by an automated data

processing machine as set forth in claim 6 including the step of analyzing the selected circuit topology at the final values of the circuit elements.

8. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

- a. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial element values,
- b. generating and storing representations of analytical derivatives of said objective function with respect to design variables,
- c. generating and storing a value of said continuous objective function which depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits, and
- d. modifying at least one of the design variables in a direction and by an amount dependent on said objective function and said representations of the analytical derivatives to improve the generated and stored value of the objective function.

9. The method of designing a circuit by an automated data processing machine as set forth in claim 8 wherein the generated and stored value of said continuous objective function depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits and a greater than, less than or equal to difference between the specified performance characteristic and the synthesized performance characteristic.

10. The method of designing a circuit by an automated data processing machine as set forth in claim 9 including repeating steps (a), (b), (c) and (d) for a preselected number of passes utilizing the design variables as modified in the previous pass to generate and store a new objective function and to further improve the generated and stored value of the objective function.

11. The method of designing a circuit by an automated data processing machine as set forth in claim 9 wherein the continuous objective function is modified by ignoring the differences between said specified performance characteristic and said synthesized performance characteristic within a selected tolerance band at selected frequencies over said range of frequencies.

12. The method of designing a circuit by an automated data processing machine as set forth in claim 9 wherein the continuous analytical objective function is modified by ignoring negative differences between said specified performance characteristic and said synthesized performance characteristic at selected frequencies over said range of frequencies.

13. The method of designing a circuit by an automated data processing machine as set forth in claim 9 wherein the continuous analytical objective function is modified by ignoring positive differences between said specified performance characteristic and said synthesized performance characteristic at selected frequencies over said range of frequencies.

14. The method of designing a circuit by an automated data processing machine as set forth in claim 9 wherein the continuous analytical objective function is modified by ignoring differences between said specified performance characteristic and said synthesized performance characteristic at selected frequencies over said range of frequencies.

15. The method of designing a circuit by an automated data processing machine as set forth in claim 8 wherein the continuous analytical objective function comprises at least one of the options selected from the following options:

$$a. f_k - (GN(f_k) - GREF(f_k))^2$$

wherein the calculated gain GN at sampled frequencies f_k is

least-squared-error fit to the a reference gain GREF:

$$b. \sum_k (GN(f_k) - GFLOAT)^2$$

wherein the calculated gain GN at sampled frequencies f_k is least-squared-error fit to a floating flat reference gain GFLOAT;

$$c. \sum_k (GFLOAT - GFLOATMAX)$$

wherein GFLOATMAX is the maximum limit of the amplitude of a floating reference gain GFLOAT and GFLOAT is maximized at sampled frequencies f_k ;

$$d. \sum_k (NF(f_k) - SPECNF)^2$$

wherein the calculated noise figure NF at sampled frequencies f_k is least-squared-error fit to a reference noise figure SPECNF;

$$e. \sum_k (MMS(f_k))^2$$

wherein the calculated source mismatch MMS at sampled frequencies f_k is least-squared-error fit to zero source mismatch;

$$f. \sum_k (MML(f_k))^2$$

wherein the calculated load mismatch MML at sampled frequencies f_k is least-squared-error fit to zero load mismatch;

$$g. \sum_k \sum_{fe} (GN(f_k + \Delta fe) - GREF(\Delta fe))^2$$

wherein the calculated gain GN at sampled frequencies $f_k + \Delta fe$, where f_k are center frequencies and Δfe define a distribution of sampling about such center frequencies, is least-squared-error fit to a reference gain GREF;

$$h. \sum_k ((\text{real}(Z(IIN, IIN)) - ZSPEC11)^2 + (\text{imaginary}(Z(IIN, IIN)) - ZSPEC11)^2 + (\text{imaginary}(Z(IOUT, IOUT)) - ZSPEC22)^2)$$

wherein the impedance matrix Z of the circuit at sampled frequencies f_k is least-squared-error fit to an impedance matrix

$$\begin{matrix} ZSPEC11 & ZSPEC12 \\ ZSPEC21 & ZSPEC22 \end{matrix}$$

where IIN is the node number assigned to the input node and IOUT is the node number assigned to the output node; and

$$i. \sum_k (PH(f_k) - PHREF(f_k) - PFLOAT - SFLOAT * f_k)^2$$

wherein the calculated phase PH at sampled frequencies f_k is least-squared-error fit to a reference phase PHREF and where PFLOAT and SFLOAT are optimization parameters.

16. The method of designing a circuit by an automated data processing machine as set forth in claim 8 wherein the continuous analytical objective function is modified by ignoring negative differences between said specified performance characteristic and said synthesized performance characteristic at selected frequencies over said range of frequencies.

17. The method of designing a circuit by an automated data processing machine as set forth in claim 8 wherein the continuous analytical objective function is modified by ignoring positive differences between said specified performance characteristic and said synthesized performance characteristic at selected frequencies over said range of frequencies.

18. The method of designing a circuit by an automated data processing machine as set forth in claim 8 wherein the continuous analytical objective function is modified by ignoring 0 differences between said specified performance characteristic and said synthesized performance characteristic at selected frequencies over said range of frequencies.

19. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

- a. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial element values,

- b. generating and storing representations of analytical derivatives of said objective function,

- c. generating and storing a value of said continuous objective function which depends on the sum, over a range of frequencies, of a power function wherein a synthesized performance characteristic may be less than, greater than, or equal to a specified performance characteristic, and

- d. modifying at least one of the design variables in a direction and by an amount dependent on said objective function and said representations of the analytical derivatives to improve the generated and stored value of the objective function.

20. The method of designing a circuit by an automated data processing machine as set forth in claim 19 including repeating steps (a), (b), (c) and (d) for a preselected number of passes utilizing the design variables as modified in the previous pass to generate and store a new objective function and to further improve the generated and stored value of the objective function.

21. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

- a. generating and storing representations of an admittance matrix of the selected circuit topology and initial element values,

- b. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial element values with said admittance matrix,

- c. generating and storing a value of said continuous objective function which depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits, and

- d. modifying at least one of the design variables in a direction by an amount dependent on said objective function to improve the generated and stored value thereof.

22. The method of circuit design by an automated data processing machine as set forth in claim 21 including:

- e. generating and storing representations of analytical derivatives with respect to the design variables of said objective function and wherein the design variables are modified in part by an amount dependent on said analytical derivatives and said objective function.

23. The method of circuit design by an automated data processing machine as set forth in claim 22 including repeating steps (a), (b), (c), (d) and (e) for a preselected number of passes using values of the design variables as modified in the previous pass to generate and store a new objective function and to further improve the generated and stored value of the new objective function.

24. The method of designing a circuit by an automated data processing machine as set forth in claim 23 wherein the generated and stored value of said continuous objective function depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits and a greater than, less than, or equal to difference between the specified performance characteristic and the synthesized performance characteristic.

25. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

- a. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial values at one of the operating modes,

- b. generating and storing a value of said continuous objective function which depends upon the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic for the selected mode and a synthesized performance characteristic for that mode beyond preset limits,
- c. repeating steps (a) and (b) for the remainder of the operating modes to generate a continuous analytical objective function and the value thereof for each mode,
- d. generating and storing a composite objective function by a summation of all the individual mode objective functions, and
- e. modifying at least one of the design variables in a direction and by an amount dependent on the composite objective function to improve the generated and stored value thereof.

26. The method of designing a circuit by an automated data processing machine as set forth in claim 25 wherein the generated and stored value of said continuous objective function depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic and a synthesized performance characteristic beyond preset limits and a greater than, less than, or equal to difference between the specified performance characteristic and the synthesized performance characteristic.

27. The method of designing a circuit by an automated data processing machine as set forth in claim 26 including repeating steps (a), (b), (c), (d) and (e) for a selected number of passes for each operating mode utilizing design variables of a previous pass to generate and store a new objective function for each mode and a new composite objective function to further improve the generated and stored value of the objective function.

28. The method of designing a circuit by an automated data processing machine as set forth in claim 27 including the step of analyzing the selected circuit topology at the final values of the circuit elements for each operating mode.

29. The method of designing a circuit by an automated data processing machine as set forth in claim 27 wherein the continuous analytical objective function comprises at least one of the options selected from the following options:

$$a. \sum f_k (GN(f_k) - GREF(f_k))^2$$

wherein the calculated gain GN at sampled frequencies f_k is least-squared-error fit to a reference gain GREF;

$$b. \sum f_k (GN(f_k) - GFLOAT)^2$$

wherein the calculated gain GN at sampled frequencies f_k is least-squared-error fit to a floating flat reference gain GFLOAT;

$$c. \sum f_k (GFLOAT - GFLOATMAX)$$

wherein GFLOATMAX is the maximum limit of the amplitude of a floating reference gain GFLOAT and GFLOAT is maximized at sampled frequencies f_k ;

$$d. \sum f_k (NF(f_k) - SPECNF)^2$$

wherein the calculated noise figure NF at sampled frequencies f_k is least-squared-error fit to a reference noise figure SPECNF;

$$e. \sum f_k (MMS(f_k))^2$$

wherein the calculated source mismatch MMS at sampled frequencies f_k is least-squared-error fit to zero source mismatch;

$$f. \sum f_k (MML(f_k))^2$$

wherein the calculated load mismatch MML at sampled frequencies f_k is least-squared-error fit to zero load mismatch;

$$g. \sum f_k \Delta fe - (GN(f_k + \Delta fe) - GREF(\Delta fe))^2$$

wherein the calculated gain GN at sampled frequencies $f_k + \Delta fe$, where f_k are center frequencies and Δfe define a distribution of sampling about such center frequencies, is least-squared-error fit to a reference gain GREF;

$$h. \sum f_k - ((\text{real}(Z(IIN, IIN)) - ZSPEC11))^2 + (\text{imaginary}(Z(IIN, IIN)) - ZSPEC11))^2 + \dots + (\text{imaginary}(Z(IOUT, IOUT)) - ZSPEC22))^2$$

wherein the impedance matrix Z of the circuit at sampled frequencies f_k is least-squared-error fit to an impedance matrix

$$\begin{matrix} ZSPEC11 & ZSPEC12 \\ ZSPEC< & ZSPEC22 \end{matrix}$$

where IIN is the node number assigned to the input node and IOUT is the node number assigned to the output node; and

$$i. \sum f_k - (PH(f_k) - PHREF(f_k) - PFLOAT - SFLOAT * f_k)^2$$

wherein the calculated phase PH at sampled frequencies f_k is least-squared-error fit to a reference phase PHREF and where PFLOAT and SFLOAT are optimization parameters.

30. The method of designing a circuit to improve the operation thereof for at least one performance characteristic by an automated data processing machine having input thereto a representative circuit topology and initial values of all design variables comprising:

a. generating and storing representations of an admittance matrix of the selected circuit topology and initial element values for one of the operating modes,

b. generating and storing a representative continuous analytical objective function for a circuit of a selected topology and initial element values with said admittance matrix for the selected operating mode,

c. generating and storing a value of said continuous objective function which depends on the sum, over a range of frequencies, of a power function of the difference between a specified performance characteristic for the selected mode and a synthesized performance characteristic for that mode beyond preset limits,

d. repeating steps (a), (b) and (c) for the remainder of the operating modes to generate a continuous analytical objective function and the value thereof for each operating mode,

e. generating and storing a composite objective function by a summation of all the individual mode objective functions,

f. modifying at least one of the design variables in a direction by an amount dependent on the composite objective function to improve the generated and stored value thereof.

31. The method of designing a circuit by an automated data processing machine as set forth in claim 30 including:

g. generating and storing representations of analytical derivatives with respect to the design variables of said objective function and wherein the design variables are modified in part by an amount dependent on said analytical derivatives and said objective function.

* * * * *